

國立成功大學
什麼什麼什麼什麼學系研究所
碩士論文

你的中文論文標題

The english title of your paper

研究生：你的中文名字
指導教授：指導教授1 指導教授2

中華民國一一零年七月

你的中文論文標題

你的中文名字* 指導教授1[†] 指導教授2[†]

國立成功大學什麼什麼什麼什麼學系

摘要

本研究之目的...

關鍵字：關鍵字、關鍵字、關鍵字、關鍵字

*學生

[†]指導教授

[†]指導教授

The english title of your paper

Author: Your name in english*

Advisor: advisor1[†] advisor2[†]

Department of what you are studying

National Cheng Kung University

SUMMARY

Write down your summary of this paper here...

Keywords: Keyword1, Keyword2, Keyword3, Keyword4

*Student

[†]Advisor

[†]Advisor

INTRODUCTION

The study on...

MATERIALS AND METHODS

There are...

RESULTS AND DISCUSSION

The results...

CONCLUSION

The topics discussed...

誌謝

在這裡寫下你的致謝詞吧。

研究生 XXX

目錄

中文摘要	I
Abstract	II
誌謝	IV
目錄	V
表目錄.....	VIII
圖目錄.....	IX
符號說明.....	X
第一章 緒論.....	1
1-1. 研究動機	1
1-2. XXXXXXXXXXXXXXXXXXXXXXXXX文獻回顧	1
1-3. 本文結構	1
第二章 文獻回顧.....	2
2-1. 語音合成技術	2
2-1.1 Tacotron	4
2-1.2 Meta-TTS	7
2-1.3 IndexTTS2	10

2-2. 「預訓練—微調」範式下之代表性模型 (Representative Models under the Pre-train, Fine-tune Paradigm)	14
2-2.1 AASIST 模型	14
2-2.2 MFA-Conformer (Conformer with Transfer Learning)	16
2-3. Prompting Paradigm	17
2-3.1 GSLM	18
2-3.2 SpeechPrompt 架構	20
2-3.3 模型比較與本計畫定位	26
第三章 實驗結果.....	28
3-1. 實驗資料集	28
3-2. 模型配置與參數設定	32
3-2.1 硬體環境與實驗控制	32
3-2.2 模型實作細節	32
3-3. 主要大實驗 Model Performance	34
3-4. Discussion/Analysis	36
第四章 前端應用.....	40
4-1. 系統架構與設計目標	40
4-2. 介面功能與模組說明	40
4-2.1 語音輸入	41
4-2.2 音訊生成與偽造	41
4-2.3 偵測核心與效能呈現	41
4-3. 後端推論流程與模型優化	42
第五章 結果及討論.....	43
5-1.	43

5-2.	43
5-2.1	43
5-3.	43
5-3.1	43
5-3.2	43
第六章 建議與未來展望.....		44
6-1.	44
6-2.	44
6-2.1	44
6-3.	44
6-3.1	44
6-3.2	44
參考文獻.....		45

表 目 錄

表 2.1 5 分制平均意見分數 (MOS) 評估[14]	6
表 2.2 Tacotron 超參數與網路架構[14]	6
表 2.3 以相似度平均意見分數 (SMOS) 評估之語者相似度（95% 信賴區間）[16]	10
表 2.4 IndexTTS2 與其他基準模型在 LibriSpeech test-clean 資料集上的效能比較[16]	13
表 2.5 不同架構之性能與參數比較表。SpeechPrompt 的參數計算基於 Prompt Length $L = 5$ 。	26
表 3.1 實驗資料集統計概覽	30
表 3.2 各模型在不同資料集與訓練設定下之效能表現	35
表 3.3 SpeechPrompt v2 在 Prompt Tuning 與 Full Fine-Tuning (Upper Bound) 效能比較	37
表 3.4 MFA-Conformer 不同預訓練模型規模之效能 (EER) 比較	38
表 3.5 MFA-Conformer 不同後端分類器架構之效能 (EER) 比較	39

圖 目 錄

圖 2.1 Tacotron 模型架構圖[14]	6
圖 2.2 Meta-TTS 訓練 MAML 流程圖[8]	9
圖 2.3 不同適應步驟下的餘弦相似度矩陣[8]	10
圖 2.4 IndexTTS2 模型架構圖[16]	11
圖 2.5 IndexTTS2 T2S 模塊架構圖[16]	11
圖 2.6 IndexTTS2 S2M 模塊架構圖[16]	12
圖 2.7 AASIST 模型架構圖	14
圖 2.8 MFA-Conformer 模型架構圖	16
圖 2.9 GSLM 架構圖	19
圖 2.10 SpeechPrompt 模型架構圖	21
圖 2.11 SpeechPrompt 在各式分類任務上的表現	25
圖 3.1 Caption	31

符號說明

a 入射波之波幅

∇ 排水體積

X

第一章 緒論

1-1. 研究動機

1-2. XXXXXXXXXXXXXXXXXXXXXXXXX文獻回顧

1-3. 本文結構

第二章 文獻回顧

2-1. 語音合成技術

目前在語音合成技術中最普遍的應用為文字轉語音（TTS, Text to Speech），其流程為輸入一段文本序列並產出相應的語音訊號。TTS 領域包含多種技術手段，從早期的單一語者端對端架構，發展至整合遷移學習（Transfer Learning）的多語者模型。近期研究則逐漸轉向以大型語言模型（LLM, Large Language Model）驅動的架構，利用海量通用語音數據提升生成效能，並根據生成機制分為逐記號產生的自回歸（AR, Auto-Regressive）與並行預測的非自回歸（non Auto-Regressive）系統。

在語音合成技術的演進過程中，傳統的統計參數語音合成管道極為複雜，通常包含文本分析前端、時長模型、聲學特徵預測模型以及基於信號處理的聲碼器等複數階段 [15]。由於各個元件需依賴大量的領域專業知識進行獨立設計與訓練，各階段產生的預測錯誤往往會不斷累積，導致最終合成音訊的自然度受限且工程開發成本高昂。為了解決上述多階段系統的侷限性，端對端（End-to-End）生成模型架構（如 Tacotron [14]）應運而生。該類模型具備從字元序列直接映射至原始頻譜圖的能力，僅需透過「文本—音訊」對即可從頭開始進行訓練，無需人工預先標註音素等級的對齊資訊。

為了提升個人化語音合成（亦稱語音複製 Voice Cloning）的實用性，利用少數的樣本來生成高品質的目標聲音已成為目標。針對此需求，學界引入了遷移學習（Transfer Learning）的技術框架。在語音領域，遷移學習通常先在大型語音數據集上訓練一個預訓練模型，隨後將此模型的知識應用於目標任務中。目前在遷移學習的架構下，構建個人化語音系統主要有兩大路徑，分別透過不同的方式處理預訓練模型的知識遷移：語者自適應（Speaker Adaptation[2, 8] 與語者編碼（Speaker

Encoding) [1, 3, 9, 16]，但兩者各具其侷限性：

- 語者自適應（Speaker Adaptation）：此方法透過少量樣本微調預訓練的多語者 TTS 模型。雖然音質通常較優，但高品質的適應往往需要數千次的微調步驟，這導致了極大的計算資源消耗與時間成本（可能長達數分鐘甚至數小時）。為了改善此效率問題並降低訓練成本，後續研究如 Meta-TTS [8] 結合了元學習（Meta-Learning）技術以實現少樣本學習（Few-shot）。元學習的核心概念在於「學習如何學習（Learn to learn）」，旨在讓模型獲得快速適應新任務的元知識，使其僅需極少量的樣本即可迅速收斂，達成高效的語者適應。
- 語者編碼（Speaker Encoding）：此方法透過在大規模多語者資料集上預訓練的語者編碼器，將輸入參照語音映射為單一語者嵌入向量（Speaker Embedding）。其最顯著的優點在於不需重新進行任何微調步驟，即可進行語音複製，實現了零樣本學習（Zero-shot），因此能提供最快的複製速度。然而，其效能高度受限於模型訓練時見過的語者資訊，當面對訓練集之外的陌生語者時，常會因為泛化落差（Generalization Gap）而導致合成聲音的相似度不如預期。

針對生成效果的評鑑，客觀指標常採用詞錯率（WER, Word Error Rate），藉由自動語音識別（ASR, Automatic Speech Recognition）技術量測合成內容與原始文本的一致性。主觀評價則依賴平均意見分數（MOS, Mean Opinion Score），由受測者針對自然度進行 1 至 5 分的評比，其中 1 分為最不自然，5 分則最接近真人。此外，為精確衡量特定聲學面向，亦採用語者相似度（SMOS, Similarity MOS）、韻律（PMOS, Prosody MOS）、音訊品質（QMOS, Quality MOS）、情緒保真度（EMOS, Emotion MOS）等衍生指標，達成對合成語音感知品質的量化。

針對語音生成效果的評鑑，常使用主觀指標平均意見分數（MOS, Mean Opinion

Score），由受測者針對自然度進行 1 至 5 分的評比，其中 1 分為最不自然，5 分則最接近真人。此外，為精確衡量特定聲學面向，亦採用語者相似度（SMOS, Similarity MOS）、韻律（PMOS, Prosody MOS）、音訊品質（QMOS, Quality MOS）、情緒保真度（EMOS, Emotion MOS）等衍生指標。在客觀評鑑方面，近期語音合成研究普遍採用詞錯率（WER, Word Error Rate）與語者相似度（Speaker Similarity, SS）作為主要指標。得益於自動語音識別（ASR）技術的突破，如 Whisper 與 FunASR 等模型已具備接近人類的辨識水準，大幅提升了 WER 自動化評測的可信度。而在語者相似度方面，則多利用語者確認模型（Speaker Verification Models）提取語音表徵（Representations），並計算合成語音與參考音訊之間的餘弦相似度（Cosine Similarity），以精確量化兩者在聲學特徵上的吻合程度。

以下將會介紹在三種種類 TTS 中經典做法：採端對端架構且針對單一語者訓練的 Tacotron [14]、結合 Meta-Learning 做在語者自適應 Few-shot TTS 的 Meta-TTS [8]、以及結合大語言模型語者編碼 Zero-shot TTS 的 IndexTTS2 [16]。

2-1.1 Tacotron

Tacotron 由 Google 的研究人員開發，是一種基於序列到序列（seq2seq）[13]的注意力機制架構，可直接從字元合成語音。如圖2.1所示，該模型的核心架構由編碼器、解碼器與後處理網路組成。編碼器的處理流程始於字元嵌入，隨即進入 Pre-net 模組。Pre-net 由帶有 Dropout 的全連接層組成，作為瓶頸層（Bottleneck Layer）以促進收斂並提升泛化能力。經過 Pre-net 處理後的特徵隨即輸入至 CBHG 模組，該模組依序結合了一維卷積濾波器組（1-D Convolutional Bank）、高速公路網路（Highway Networks）[12] 與雙向閘控循環單元（bidirectional gated recurrent unit (GRU)[4]）。此設計旨在捕捉類似 N-gram 的局部特徵以及全域的雙向上下文資訊。隨後的解碼階

段採用內容導向的 tanh 注意力機制，並透過縮減因子（Reduction Factor）在每個時間步預測多個頻譜幀。

在特徵生成與音訊重建方面，解碼器輸出的 80 階梅爾頻譜圖會經由後處理網路進一步轉化為線性頻率規模頻譜圖。此階段再次運用 CBHG 模組的雙向處理能力，修正幀級別的預測誤差，強化諧波結構。最終，該系統採用 Griffin-Lim 演算法從線性頻譜圖中合成音訊波形。實驗數據顯示，在美國英語的平均意見分數（MOS）測試中，此模型獲得 3.82 的評分，超越了傳統的生產級參數化合成系統。此外，由於該模型是在幀級別而非樣本級別生成語音，其推理速度明顯優於如 WaveNet 等自迴歸模型。

訓練過程中，該架構直接在文字與音訊配對的原始資料上運行，無需人工標註音素級別的對齊。損失函數採用對解碼器與後處理網路輸出的 L1 損失進行同等加權計算，並透過 Adam 優化器進行訓練。研究指出，Pre-net 中的 Dropout 對於模型在缺乏計畫採樣（Scheduled Sampling）的情況下達成泛化至關重要。消歧義實驗證實，若缺乏 CBHG 或後處理網路，模型的對齊能力與音質解析度將顯著下降。該研究為端對端語音合成奠定了基礎，展現了簡化流水線在應對真實世界噪音數據與適應多樣語音屬性方面的潛力。

評估語音合成的自然度，作者進行了 MOS 測試，測試母語者對 100 個未再訓練中見過的短語，每個短語收集 8 個評分，並且在計算 MOS 時僅計入使用耳機的評分數據。實驗將 Tacotron 與基於 LSTM 的參數式系統（Parametric）[15] 以及拼接式系統（Concatenative）[7] 比較。如表 2.1 所示，Tacotron 達到了 3.82 的 MOS 分數，成功超越了參數式系統。



圖 2.1: Tacotron 模型架構圖[14]

系統	MOS
Tacotron	3.82 ± 0.085
參數式 (Parametric)[15]	3.69 ± 0.109
拼接式 (Concatenative)[7]	4.09 ± 0.119

表 2.1: 5 分制平均意見分數 (MOS) 評估[14]

Spectral analysis	pre-emphasis: 0.97; frame length: 50 ms; frame shift: 12.5 ms; window type: Hann
Char embedding	256-D
Encoder CBHG	<u>Conv1D bank</u> : $K=16$, conv- k -128-ReLU <u>Max pooling</u> : stride=1, width=2 <u>Conv1D projections</u> : conv-3-128-ReLU → conv-3-128-Linear <u>Highway net</u> : 4 layers of FC-128-ReLU <u>Bidirectional GRU</u> : 128 cells
Encoder pre-net	FC-256-ReLU → Dropout(0.5) → FC-128-ReLU → Dropout(0.5)
Decoder pre-net	FC-256-ReLU → Dropout(0.5) → FC-128-ReLU → Dropout(0.5)
Decoder RNN	2-layer residual GRU (256 cells)
Attention RNN	1-layer GRU (256 cells)
Post-processing	<u>Conv1D bank</u> : $K=8$, conv- k -128-ReLU <u>Max pooling</u> : stride=1, width=2 <u>Conv1D projections</u> : conv-3-256-ReLU → conv-3-80-Linear <u>Highway net</u> : 4 layers of FC-128-ReLU <u>Bidirectional GRU</u> : 128 cells
Reduc. factor (r)	2

表 2.2: Tacotron 超參數與網路架構[14]

2-1.2 Meta-TTS

Meta-TTS 旨在解決傳統文字轉語音模型在適應新說話者時，往往需要大量數據與訓練步驟的挑戰。傳統微調方法通常需要數千步才能獲得高品質結果，不僅速度較慢且容易產生過擬合現象。為了實現少樣本語音複製並加速適應過程，該研究提出了 Meta-TTS 架構，將「與模型無關的元學習算法（MAML）[5]」應用於非自回歸模型 FastSpeech 2 之上。透過此結合，模型能學習到一組具備快速適應能力的元初始化參數，使其在面對未見過的新說話者時，僅需極少量的樣本與幾次梯度下降步驟，即可達到良好的合成效能。本節後續將首先闡述其基底架構 FastSpeech 2 的運作原理，接著詳述元學習演算法如何整合應用於該模型之中。

FastSpeech 2 [11]為一種非自回歸的文字轉語音模型，主要由編碼器（Encoder）、變異適配器（Variance Adapter）與梅爾頻譜解碼器（Mel-spectrogram Decoder）組成。編碼器由 Transformer 層堆疊而成，負責將音素嵌入序列轉換為隱藏序列以提取上下文；變異適配器負責加入時長、音高與能量等資訊以決定語速與語調；解碼器則將適配後的序列並行轉換為決定音色的梅爾頻譜序列。在多語者版本中，模型將目標語者的嵌入向量（Speaker Embedding）加入變異適配器與解碼器的輸入作為條件，使這兩個模組能根據特定特徵進行生成，而編碼器因僅處理文本內容故不加入說話者資訊。當針對新說話者進行微調時，流程通常會固定編碼器參數，僅利用少量樣本微調語者嵌入以及變異適配器與解碼器，以適應新的聲音特徵。

由於多語者語音合成系統可視為單一語者合成的多任務版本，因此可透過微調將預訓練的多語者模型適應至新說話者。在語者自適應方法中，常見的微調策略主要有兩種：一是僅微調語者嵌入 $\{E_S\}$ ，二是微調整個模型 $\{\theta_E, \theta_{VA}, \theta_D, E_S\}$ 。然而，由於原始的語者嵌入查找表 E_S 僅包含訓練集中的說話者資訊，對於測試集中

的每位新說話者 i ，必須初始化一個新的嵌入表 \hat{E}_S 以獲得對應的嵌入向量 \hat{e}_i 。此外，鑑於編碼器 θ_E 不應受說話者身分制約，微調階段通常不會對其進行更新。綜合考量下，該研究的實驗主要聚焦於同時微調變異適配器、解碼器與新語者嵌入 $\{\theta_{VA}, \theta_D, \hat{E}_S\}$ ，以在適應效率與模型效能間取得平衡。

元學習（Meta-learning）又被稱為「學習如何學習（learn to learn）」，其目標在於設計出能利用少量訓練樣本快速適應新環境或學習新技能的模型，因此極為適合應用於少樣本下游任務。與傳統監督式學習針對單一資料集進行擬合不同，元學習係透過在訓練任務集上的擬合，學習出一組良好的模型元初始化參數（Meta-initialization），以利於後續的遷移學習。該研究選用 MAML[5] 演算法作為核心，並針對語音合成特性進行了架構調整。具體而言，該研究參考了「幾乎無內迴圈」（Almost No Inner Loop, ANIL）[10]的概念，但其基本理念有所不同。ANIL 基於特徵重用（Feature Reuse）的觀察，僅在內迴圈更新最後一層參數；然而，該研究指出在 Meta-TTS 中並未觀察到顯著的特徵重用現象，而是根據模組職責進行區分。鑑於變異適配器、解碼器與語者嵌入對說話者適應的影響較為顯著，該研究設計在內迴圈中僅針對這些與說話者高度相關的模組進行梯度更新，而固定編碼器參數，隨後在外迴圈中才對整體模型進行元更新。

圖 2.2 該圖左半部表示內迴圈之前的初始模型參數 θ ，而右半部則表示經過內迴圈梯度下降後的自適應模型參數 θ_i 。圖中紅色虛線箭頭與紅色方塊明確標示出哪些參數（即變異適配器、解碼器與語者嵌入）在內迴圈過程中進行了更新。在完成內迴圈適應後，系統將查詢集的輸入資料前饋至右側已適應的模型中，並計算其輸出損失作為該元任務的損失。最終，該損失將依循綠色虛線箭頭的路徑，從圖的右上角反向傳播回圖左半部的初始模型參數，以完成外迴圈的參數優化。

為了驗證 Meta-TTS 在少樣本語者適應任務上的有效性，該研究使用了 LibriTTS 的 train-clean-100 子集進行模型訓練，並在 LibriTTS test-clean 與 VCTK 資料集上進

行測試。實驗中的基準模型（Baseline）指未經元學習訓練、僅透過傳統微調進行適應的多語者 FastSpeech 2 模型。在主觀評估方面，實驗集中於 10 個適應步驟（Adaptation steps）的情境下測量相似度平均意見分數（SMOS）。實驗中使用該聲碼器將真實梅爾頻譜圖轉換為參考基線（稱為重構語音），並將其評測結果視為合成語音相似度的理論上限。表 2.3 的結果顯示，無論是在同源的 LibriTTS 或跨源的 VCTK 測試中，Meta-TTS 的表現皆顯著優於基準模型，SMOS 分數差距超過 1 分。在客觀評估方面，圖 2.3 展示了不同適應步驟下的餘弦相似度矩陣。觀察矩陣變化可知，基準模型約需 50 個適應步驟才能顯現出與目標語者相似的對角線模式，而 Meta-TTS 僅需 5 至 10 個步驟即可達到同等清晰度，證實其在極少量的更新步驟下即可快速生成高相似度的目標語音。

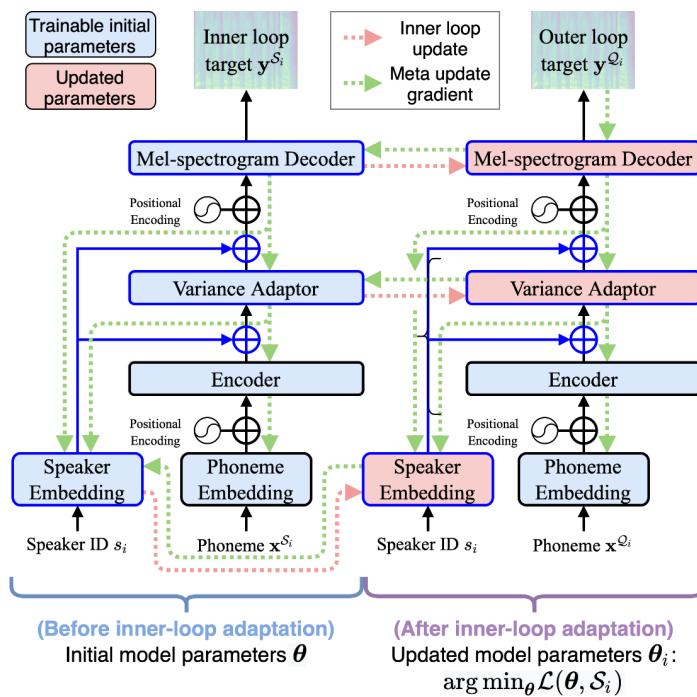


圖 2.2: Meta-TTS 訓練 MAML 流程圖[8]



圖 2.3: 不同適應步驟下的餘弦相似度矩陣[8]

Approach	Adaptation	LibriTTS			VCTK		
		Emb table \hat{E}_S	Shared e_S	Emb table \hat{E}_S	Shared e_S		
Real			4.29 ± 0.27			4.54 ± 0.09	
Reconstructed			3.33 ± 0.29			4.08 ± 0.12	
Baseline	10 steps	1.53 ± 0.18	1.34 ± 0.21	1.56 ± 0.12	1.32 ± 0.13		
Meta-TTS	10 steps	2.77 ± 0.24	2.67 ± 0.28	3.14 ± 0.16	3.45 ± 0.14		

表 2.3: 以相似度平均意見分數 (SMOS) 評估之語者相似度 (95% 信賴區間) [16]

2-1.3 IndexTTS2

IndexTTS2 是由 bilibili 人工智慧平台部門提出的一種大型自回歸零樣本語音合成模型。當前語音合成技術已顯著轉向以大型語言模型（LLM）驅動的開發路徑，隨著如 XTTS [1]、F5-TTS [3] 與 Fish-Speech [9] 等系統透過大量語音記號（speech tokens）與大規模數據訓練，系統得以在複雜的潛在空間（Latent space）中穩健捕捉音色（Timbre）與韻律（Prosody）。此類進展有效解決了以往編碼器面對陌生講者時相似度不足的痛點。IndexTTS2 的核心創新在於解決了自回歸（AR）模型對時間

掌控的困難，提出了一種既能精確控制時長，又能將語者身分與情感表達獨立控制的架構。



圖 2.4: IndexTTS2 模型架構圖[16]

該模型採用級聯式架構，由文字轉語意（Text-to-Semantic, T2S）、語意轉梅爾頻譜（Semantic-to-Mel, S2M）以及聲碼器（Vocoder）三個核心模塊組成。T2S 模塊負責從文本生成語意記號，S2M 模塊將這些記號轉換為梅爾頻譜圖，最後由 BigVGANv2 聲碼器將頻譜圖轉換為音訊波形。為了增強情緒表達的穩定性與清晰度，該研究設計了一種三階段訓練策略。第一階段使用全量數據訓練基礎模型並隨機歸零時長嵌入以支援自由生成；第二階段引入梯度反轉層（GRL, Gradient Reversal Layer）[6]與情緒適配器，專注於情緒特徵與語者特徵的解耦；第三階段則凍結所有特徵調節器，對全模型進行微調以提升魯棒性。



圖 2.5: IndexTTS2 T2S 模塊架構圖[16]

在文字轉語意（T2S）模塊中，模型採用自回歸 Transformer 架構，其輸入序列定義為 $[c, p, e_{(BT)}, E_{text}, e_{(BA)}, E_{sem}]$ ，其中 c 代表語者屬性， p 為時長控制嵌入， E_{text} 與 E_{sem} 分別為文本與語意記號的嵌入向量。為了降低情緒控制的門檻，該研究引入了文字轉情緒（Text-to-Emotion, T2E）模塊，利用知識蒸餾技術將 Deepseek-r1 的推論能力轉移至參數量較小的 Qwen-3-1.7b 模型。該模塊能將自然語言描述映射為情緒機率分布，進而生成情緒向量 e 供 T2S 使用。在時長控制方面，透過約束時長嵌入表 W_{num} 與語意位置嵌入表 W_{sem} 相等（即 $W_{sem} = W_{num}$ ），模型能精確將位置資訊與目標時長 T 對齊。



圖 2.6: IndexTTS2 S2M 模塊架構圖[16]

語意轉梅爾頻譜（S2M）模塊則採用基於流匹配（Flow Matching）的非回歸架構，該架構通過學習常微分方程（ODE）將雜訊分布映射至目標梅爾頻譜。研究團隊發現，在合成高強度的情緒語音時，僅依賴語意記號可能導致發音含糊不清（slurring）的問題。為解決此問題，該模塊引入了 GPT 隱含特徵增強機制，將 T2S 模塊最後一層 Transformer 的輸出 H_{GPT} 與語意特徵進行融合。由於 H_{GPT} 包含豐富

的文本與上下文資訊，這種融合顯著提升了情緒表達下的語音清晰度與穩定性。

為了驗證 IndexTTS2 的效能，該研究在 LibriSpeech test-clean 等公開資料集上進行了評估，並與 MaskGCT、F5-TTS、CosyVoice2、SparkTTS、IndexTTS 等先進的零樣本學習基準模型進行對比。實驗結果顯示（如表 2.4），IndexTTS2 在語者相似度（SS）等客觀指標上達到了 0.870，而在 LibriSpeech 測試集的主觀聽測實驗（MOS）中，IndexTTS2 的表現均優於其他所有基準模型。消融實驗進一步證實，若移除 GPT 隱含特徵增強，字錯率（WER）將從 3.115% 上升至 3.334%，顯示該機制對維持發音清晰度的重要性。

Model	SS ↑	WER (%) ↓	SMOS ↑	PMOS ↑	QMOS ↑
Ground Truth	0.833	3.405	4.02 ± 0.22	3.85 ± 0.26	4.23 ± 0.12
MaskGCT	0.790	7.759	4.12 ± 0.09	3.98 ± 0.11	4.19 ± 0.19
F5-TTS	0.821	8.044	4.08 ± 0.21	3.73 ± 0.27	4.12 ± 0.13
CosyVoice2	0.843	5.999	4.02 ± 0.22	4.04 ± 0.28	4.17 ± 0.25
SparkTTS	0.756	8.843	4.06 ± 0.20	3.94 ± 0.21	4.15 ± 0.16
IndexTTS	0.819	3.436	4.23 ± 0.14	4.02 ± 0.18	4.29 ± 0.22
IndexTTS2	0.870	3.115	4.44 ± 0.12	4.12 ± 0.17	4.29 ± 0.14
- GPT latent	0.887	3.334	4.33 ± 0.10	4.10 ± 0.12	4.17 ± 0.22

表 2.4: IndexTTS2 與其他基準模型在 LibriSpeech test-clean 資料集上的效能比較[16]



圖 2.7: AASIST 模型架構圖

2-2. 「預訓練—微調」範式下之代表性模型 (Representative Models under the Pre-train, Fine-tune Paradigm)

本節將回顧目前在傳統「預訓練—微調（Pre-train, Fine-tune）」範式下在語音深偽偵測任務上的代表性模型。在此範式下，普遍以透過在大規模資料配合自監督學習方式預訓練的模型作為前端模型，串接為了特定下游任務設計的模型結構，在整體上進行參數微調。此範式在各式下游任務上建立了性能標竿，但有著需要龐大存儲空間以及算力資源的短板。以下將詳細介紹兩個在深偽音訊偵測上以及語者驗證（ASV）任務上具代表性的架構：基於圖神經網路的 AASIST，以及基於Conformer架構與多尺度特徵融合（MFA）的MFA-Conformer。

2-2.1 AASIST模型

AASIST架構是由Tak et al. (2022)所提出的，在ASVspoof21 LA挑戰中取得優異的成績。該作者在後續研究中將前端模型替換為wav2vec 2.0 XLSR並加上Rawboost Data Augmentation後被視為語音深偽偵測任務的SOTA模型。(許多後續研究以及挑戰基於此模型進行改進語比較，足見此架構之經典)

該模型的運作流程如下（架構如圖X所示）：

1. 前端特徵提取：使用預訓練的 wav2vec 2.0 模型對原始音波進行處理。由於 wav2vec 2.0 在大量語音數據上進行了自監督訓練，其提取的特徵具備高度的泛化能力與豐富的聲學細節。
2. 深層編碼：將前端輸出輸入至基於 RawNet2 的殘差編碼器，進一步學習深層的聲學特徵。
3. 特徵聚合與圖建模：透過基於自注意力的聚合層，分別提取頻譜（Spectral）和時序（Temporal）兩種維度的特徵。這兩組特徵隨後各自通過由圖注意網路（GAT）和圖池化（Graph Pooling）組成的圖模塊。
4. 異質圖融合：利用圖結合技術，將頻譜與時序特徵融合為異質譜時圖（Heterogeneous Spectro-Temporal Graph）。隨後，特徵被送入由 **HS-GAL（Heterogeneous Stacking Graph Attention Layer）** 組成的 **MGO（Max Graph Operation）** 模塊，以捕捉跨域的偽造痕跡。
5. 分類決策：最後，將 MGO 產出的兩組節點堆疊並最大化，通過各節點的最大化和平均運算後串接隱藏全連接層，輸出真偽標籤。

此架構透過全參數微調（Full Fine-tuning）wav2vec 2.0 前端與 AASIST 後端，在 ASVspoof 2021 評測中取得了優異成績，成為該挑戰下的里程碑。此模型的一個主要特徵是其前端模型 wav2vec 2.0 XLS-R 本身擁有約 317M 的巨大參數量，因此在面對未見過的攻擊類型或跨資料集測試（如 ASVspoof 2021 DeepFake 數據集）時，展現了傑出的泛化能力。然而，這也意味著其訓練與推論都需要龐大的計算資源與記憶體空間。



圖 2.8: MFA-Conformer 模型架構圖

2-2-2 MFA-Conformer (Conformer with Transfer Learning)

另一個值得關注的 SOTA 模型是 MFA-Conformer，該架構由 Zhang 等人（2022）首先應用於聲紋識別（ASV），隨後被證實同樣適用於偽造語音偵測。Conformer 區塊（Conformer Block）的設計初衷在於解決 Transformer 雖擅長捕捉長距離全域依賴（Global Context），卻在提取細微局部特徵（Local Features）上不如卷積神經網路（CNN）的問題。Conformer 透過將卷積模組（Convolution Module）嵌入 Transformer 的自注意力機制與前饋網路之間，成功結合了 CNN 的局部感知能力與 Transformer 的全域建模能力。此架構的優勢最早由 Gulati 等人（2020）[?] 在自動語音識別（ASR）任務中獲得證實：實驗顯示，在 LibriSpeech 基準測試上，Conformer 能以僅 Transformer 四分之一不到的參數量，達到更優異的辨識率（WER）。MFA-Conformer 在『參數效率（Parameter Efficiency）』上所展現的潛力，不僅證實了輕量化設計的可行性，更為本文後續進行跨範式（Cross-paradigm）的模型效能評比，提供了極具代表性的對照組。

(該說一下Conformer block的數學或是放一張Conformer block圖)

為了有效捕捉不同層級的特徵，此模型利用多尺度特徵聚合（MFA）與池化機制將所有 Conformer 區塊的輸出進行融合，而非僅使用最後一層。其具體運作流程與 ASP 數學定義如下：

第一步：特徵拼接與正規化：假設模型共有 L 個 Conformer 區塊（本研究中

$L = 16$)，第 i 層的輸出特徵序列為 H_i 。首先，將每一個 Conformer 區塊的輸出在特徵維度上進行拼接 (Concatenation)，並通過層正規化 (Layer Normalization)：

$$H_{concat} = \text{LayerNorm}(\text{Concat}(H_1, H_2, \dots, H_L)) \quad (2.1)$$

第二步：注意力統計池化 (Attentive Statistic Pooling, ASP)：為了從變長的語音序列中提取固定維度的嵌入向量 (Embedding)，拼接後的特徵 H_{concat} 會輸入至 ASP 層。假設 h_t 為 H_{concat} 在時間步 t 的特徵向量，ASP 首先透過注意力機制計算每個時間步的權重 α_t ：

$$e_t = v^T \tanh(W h_t + b) \quad (2.2)$$

$$\alpha_t = \frac{\exp(e_t)}{\sum_\tau \exp(e_\tau)} \quad (2.3)$$

其中 W 與 b 為可學習的權重與偏差， v^T 為投影向量。接著，利用權重 α_t 計算加權平均數 (Weighted Mean, $\tilde{\mu}$) 與加權標準差 (Weighted Standard Deviation, $\tilde{\sigma}$)：

$$\tilde{\mu} = \sum_t \alpha_t h_t \quad (2.4)$$

$$\tilde{\sigma} = \sqrt{\sum_t \alpha_t (h_t - \tilde{\mu})^2} \quad (2.5)$$

最後，將統計量 $\tilde{\mu}$ 與 $\tilde{\sigma}$ 進行拼接，作為最終的語句層級特徵向量 H_{MFA} ，再經由 Batch Norm 與線性層進行真偽分類：

$$H_{MFA} = \text{Concat}(\tilde{\mu}, \tilde{\sigma}) \quad (2.6)$$

2-3. Prompting Paradigm

隨著大規模預訓練模型在語音領域的廣泛應用，傳統的「預訓練—微調 (Pre-train, Fine-tune)」範式面臨了嚴重的擴展性挑戰。在此範式下，模型首先在

海量無標註數據上進行預訓練以學習通用表徵，隨後針對特定下游任務，需更新並儲存模型內部的全部或大部分參數。然而，隨著模型參數規模邁入數億級甚至更大的量級，為每個任務維護獨立的模型副本，在存儲空間與計算資源上均造成了沉重負擔。

為了解決此問題，Chang 等人 (2023, 2024) 借鑑了自然語言處理 (NLP) 領域的技術，提出了一種基於提示學習（Prompting）範式的參數高效（Parameter-Efficient）SpeechPrompt 架構。該架構的核心在於凍結預訓練語音語言模型的參數，僅透過插入少量可學習的提示向量（Prompt Vectors），將各類語音任務統一重塑為「語音到單元生成（Speech-to-Unit Generation）」任務，從而實現高效的跨任務遷移。下節將介紹 SpeechPrompt 所基於的核心技術：生成式語音語言模型 GSLM，並在後續章節介紹 SpeechPrompt 的實踐方法。

2-3-1 GSLM

傳統語音處理模型往往依賴大量標註數據，然而 Lakhota 等人 (2021) 提出的 GSLM 旨在模擬人類早期的語言習得過程，僅透過原始音訊輸入即可學習聲學與語言特徵，實現無文本的自然語言處理（Textless NLP）。

GSLM 的核心概念是將連續的語音波形離散化，並在此基礎上訓練語言模型。其標準流程包含三個階段：

1. Speech-to-Unit (S2u)：利用預訓練的自監督模型（Self-Supervised Learning, SSL）提取語音特徵，並透過 K-means 分群將連續特徵向量量化為離散單元序列（Discrete Units）。
2. Unit-based Language Model (uLM)：在離散單元上訓練 Transformer Decoder，學習單元間的機率分佈。



圖 2.9: GSLM 架構圖

3. Unit-to-Speech (u2S)：將生成的單元序列輸入解碼器，轉回連續的語音波形。

GSLM 的語言模型（uLM）採用標準的 Transformer 架構，並以自回歸（Autoregressive）方式進行預訓練。其訓練目標是最大化下一個離散單元的對數似然函數（Log-Likelihood）。

假設一段語音被編碼為長度為 T 的離散單元序列 $u = (u_1, u_2, \dots, u_T)$ ，則模型的訓練損失函數 \mathcal{L} 定義為：

$$\mathcal{L} = - \sum_{t=1}^T \log P(u_t | u_{<t}; \theta) \quad (2.7)$$

其中 $u_{<t}$ 表示當前時刻之前的所有單元 (u_1, \dots, u_{t-1}) ， θ 為模型參數。透過最小化此損失函數，模型能夠捕捉語音中長距離的依賴關係與語法結構，建立語言的機率模型。

在訓練數據方面，Lakhotia 等人（2021）使用了 LibriLight 資料集中的 6k 小時無標註「乾淨（clean）」語音進行訓練。這證實了 GSLM 具備從大規模原始音訊中歸納高階語言知識的能力，而無需依賴任何人工轉寫的文本資源。

GSLM 的生成品質高度依賴於前端 S2u 編碼器的特徵提取能力。Lakhotia 等人（2021）系統性地比較了 CPC、wav2vec 2.0 與 HuBERT 三種主流的自監督模型。研究發現，不同的編碼器對下游生成的影響顯著不同。其中，HuBERT 在語音重合成（Resynthesis）與生成（Generation）的各項客觀指標（如 Perplexity）與主觀指標（如人類聽測 MOS）上，往往能取得較佳的平衡，特別是在捕捉語音內容（Content）方面表現穩健。基於文獻中的發現，後續的相關研究（如 SpeechPrompt）傾向於選擇 HuBERT 作為默認的特徵提取器。

2-3-2 SpeechPrompt 架構

SpeechPrompt的核心概念是將預訓練的語音語言模型（如 GSLM）視為凍結(Frozen)的黑盒，僅透過在輸入端或模型內部層級中注入極少量（通常小於總量0.1%）的可學習「提示向量(Prompt Vectors)」，來引導(Prompting)模型利用其預訓練所獲得的知識來處理多樣化的下游任務。

為了更清晰地闡述 SpeechPrompt 的運作機制，我們將架構圖中的關鍵模組進行細部分解與定義。這些模組共同協作，將輸入語音轉換為下游任務的預測結果：

- SSL Quantizer(自監督語音量化器)：作為系統的輸入前端，此模組負責將連續的語音波形轉換為離散單元序列（Token Sequence）。如 2.1.3 節所述，本研究考量其特徵提取的穩健性，選用預訓練的 HuBERT 模型提取特徵，並透過 K-means 演算法進行量化。這一步驟是將連續訊號「語言化」的關鍵。
- Prompt Vectors(提示向量)：這是 SpeechPrompt 架構中核心的可訓練參數，負責攜帶任務特定的指令資訊。其主要包含前置於輸入層的 Input Prompts 以及嵌入於模型內部的 Deep Prompts。關於這些向量的具體數學定義、維度設定以及在 Attention 機制中的嵌入方式，我們將在 **2.2.3 節** 中進行詳細的形式化描



圖 2.10: SpeechPrompt 模型架構圖

述。

- Frozen Unit-based Language Model (凍結的單元語言模型)：此為整體架構的骨幹（Backbone），以下將簡稱為 uLM，負責理解輸入的語音單元並生成上下文表示（Contextual Representations）。文獻中已探討了多種模型架構，如 GSLM、pGSLM 與 Unit mBART。本計畫選用 GSLM 作為骨幹。在 Prompt Tuning 過程中，uLM 的所有參數保持凍結（Frozen），僅負責推論。這保證了下游任務訓練不會發生「災難性遺忘（Catastrophic Forgetting）」，並大幅降低了運算成本。
- Verbalizer (標籤映射器)：這是針對分類任務（如 ASVspoof）的關鍵組件，負責將語言模型輸出的高維度單元分佈（Vocabulary Distribution）映射到低維度的任務標籤空間（Label Space）。常見的選擇包括：

- Fixed Verbalizer (固定映射)：手動或隨機指定特定單元 (Units) 代表目標類別（例如指定單元 #10 代表「真實語音」）。此方法高度依賴人工設計，且要求模型必須透過原有的 uLM 嵌入表 (Embedding Table) 精確輸出特定的任務標記 (Tokens)，在性能與靈活性上通常較為受限。
- Learnable Verbalizer (可學習映射)：這是一個取代原有嵌入層映射的輕量級線性投影矩陣 (Linear Projection)，能自動學習並識別哪些單元組合最能代表特定的分類標籤。SpeechPrompt v2 的實驗證實，此方法在多數任務下皆能顯著提升分類的準確度。

SpeechPrompt 採用了結合 Input Prompt Tuning 與 Deep Prompt Tuning 的策略，在模型的不同層級注入資訊：

- Input Prompt Tuning (輸入端提示微調)

最直觀的 Prompt 嵌入方式是在模型的輸入端加入可訓練的向量。假設輸入語音經過 S2u 編碼後的離散單元序列嵌入 (Embedding) 為 $E(u) = [e(u_1), e(u_2), \dots, e(u_T)]$ 。我們定義一組可訓練的 Prompt 向量 $P^I = [p_1^I, p_2^I, \dots, p_L^I]$ ，其中 L 為 Prompt 的長度。

在輸入層，我們將 Prompt 向量序列前置於語音單元序列之前，形成新的輸入序列 X ：

$$X = \text{Concat}(P^I, E(u)) \quad (2.8)$$

透過此機制，模型在處理語音特徵序列之前，即受到任務特定參數的制約。這些前置向量提供了可學習的上下文資訊 (Contextual Information)，有效地調節模型對後續輸入的特徵提取與生成行為，使其適應特定的下游任務。

- Deep Prompt Tuning (深層提示微調)

為了增強 Prompt 對模型的控制力，SpeechPrompt 進一步採用了 Deep Prompt Tuning 技術。不同於僅在輸入層加入 Prompt，此方法在 Transformer 的每一層 Attention 機制中都嵌入了可訓練的向量。

假設 Transformer 第 j 層的輸入為隱藏狀態 h ，標準的 Self-Attention 機制計算如下：

$$Attn(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.9)$$

其中 Q, K, V 分別為 Query, Key, Value 矩陣。在 Deep Prompt Tuning 中，我們引入兩組可訓練的 Prompt 向量 P_K 與 P_V ，並將其同樣前置於原始的 Key 和 Value 矩陣之前：

$$K' = \text{Concat}(P_K, h)W_K \quad (2.10)$$

$$V' = \text{Concat}(P_V, h)W_V \quad (2.11)$$

透過這種方式，Prompt 向量 P_K 與 P_V 能夠直接影響每一層 Attention 的權重分佈與輸出值。這意味著我們僅需訓練極少量的參數（即 P^I, P_K, P_V ），就能在不改變模型權重的情況下，有效地「重新程式化（Reprogramming）」模型的行為以適應下游任務。

SpeechPrompt 架構根據下游任務的性質，採用不同的訓練目標。針對語音分類任務（如 ASVspoof），模型輸出的離散單元機率分佈會通過 Learnable Verbalizer 投

影到類別空間，並透過最小化交叉熵損失函數（Cross-Entropy Loss）來優化 Prompt 向量與 Verbalizer 參數：

$$\mathcal{L}_{CE} = - \sum_{c=1}^C y_c \log(\hat{y}_c) \quad (2.12)$$

其中 C 為類別總數。對於序列生成任務（如 ASR），模型則採用自回歸方式最大化目標序列的條件機率。其損失函數定義為：

$$\mathcal{L}_{Gen} = - \sum_{t=1}^M \log P(y_t | y_{<t}, X; \theta_{prompt}) \quad (2.13)$$

在此過程中，模型透過最大化似然函數學習如何根據 Prompt 上下文生成符合目標的單元序列。

Chang 等人（2023, 2024）的研究展示了 SpeechPrompt 架構的廣泛適用性。除了本計畫關注的語音分類任務外，此架構亦透過將任務轉化為序列生成問題，成功應用於自動語音辨識 (ASR)、語音翻譯 (ST) 與語音延續 (Speech Continuation) 等生成式任務，證實了 Prompting 典範作為通用語音處理框架的靈活性與潛力。

針對語音分類領域，Chang 等人（2023）在 SpeechPrompt v2 的研究中進行了廣泛評估。為了驗證此架構的潛力與適用邊界，我們引用其實驗數據（如圖三所示）進行詳細分析。

根據圖三的實驗數據，我們可以將 SpeechPrompt v2 在不同任務上的表現歸納為三個層次，這證明了 SpeechPrompt 是一個具備高度前景的通用架構，但也存在特定的改進空間：

1. 超越 SOTA (Outperform SOTA)：在某些任務中，SpeechPrompt 甚至擊敗了全參數微調的專用模型，展現了 GSLM 在語意理解上的強大優勢。
 - 具體任務：立陶宛語關鍵詞偵測 (Lithuanian SCR)、阿拉伯語關鍵詞偵測 (Arabic SCR)、以及諷刺偵測 (Sarcasm Detection)。

Task	Metric	Dataset	Language	#Class	SOTA (Topline)	GSLM	GSLM+	pGSLM	pGSLM+
SCR	ACC (↑)	Google SC v1	En	12	98.6 [10]	94.5	94.6	94.3	94.7 (-3.9)
		Grabo SC	Du	36	98.9 [11]	92.4	92.7 (-6.2)	17.5	19.6
		Lithuanian SC	Lt	15	91.8 [9]	93.2	95.5 (+3.7)	90.9	79.5
		Arabic SC	Ar	16	98.9 [9]	99.7	100.0 (+1.1)	85.6	92.6
IC	ACC (↑)	Fluent SC	En	24	99.7 [12]	97.2	97.3	98.1	98.2 (-1.5)
LID	ACC (↑)	Voxforge	En, Es, Fr De, Ru, It	6	99.8 [13]	90.9	94.2 (-5.6)	81.8	80.4
FSD	EER (↓)	ASVspoof	En	2	2.5 [13]	18.5	13.5	13.1 (+10.6)	18.3
ER	ACC (↑)	IEMOCAP	En	4	79.2 [13]	42.1	44.3	49.9	50.2 (-29)
AcC	ACC (↑)	AccentDB	En	9	99.5 [14]	78.9	83.4	86.5	87.1 (-12.4)
SD	F1 (↑)	MUSTARD	En	2	64.6 [15]	55.0	77.8	74.4	78.7 (+13.1)
		MUSTARD++	En	2	65.2 [16]	74.0	75.2 (+10)	52.7	58.2
GID	F1 (↑)	VoxCeleb1	En	2	98.3 [17]	86.2	87.3	91.6 (-6.7)	86.2
VAD	ACC (↑)	Google SC v2 & Freesound	En	2	98.8 [18]	96.6	96.9	98.3 (-0.5)	98.1
AuC	ACC (↑)	ESC-50	✗	50	97.0 [19]	9.0	37.5 (-59.5)	20.3	27.0

圖 2.11: SpeechPrompt 在各式分類任務上的表現

- 意涵：這顯示 GSLM 預訓練所學到的高階語言特徵，對於語意理解與低資源語言的適應性極佳。Prompting 技術能夠有效激發這些潛在知識，在少量數據下達成超越傳統方法的表現。

2. 比肩 SOTA (Competitive with SOTA)：在主流的標準數據集上，SpeechPrompt 僅需訓練極少量的參數（通常小於模型總參數的 0.1%），即可達到與訓練所有參數 (Fine-tuning) 相當的水準。

- 具體任務：Google Speech Commands (SCR)、語者意圖分類 (Intent Classification)、語言辨識 (LID)、性別識別 (GID) 及語音活動偵測 (VAD)。
- 意涵：這證明了 SpeechPrompt 是一個真實可行且高效的解決方案。它大幅降低了儲存與計算成本，卻未顯著犧牲性能，具備極高的實用價值與參數效率 (Parameter Efficiency)。

3. 落後 SOTA (Underperform SOTA)：在涉及細微聲學特徵或非語意內容的任務上，SpeechPrompt 與專用模型仍有明顯差距。

- 具體任務：偽造語音偵測 (Fake Speech Detection / ASVspoof)、情緒辨識 (Emotion Recognition)、口音分類 (Accent Classification)。
- 關鍵數據：特別是在本計畫關注的 ASVspoof 2019 LA (Logical Access) 任務上，當時SOTA 模型的等錯誤率 (EER) 可達 2.5%，而 SpeechPrompt v2 最佳僅能達到 13.5%。
- 意涵：這顯示出目前的 SpeechPrompt 架構在捕捉「非語意」的聲學細節（如合成語音的偽造痕跡或情緒的細微變化）時仍有局限。這可能歸因於 GSLM 前端的離散化過程（Quantization）過濾掉了部分關鍵的高頻聲學訊息，或是目前的 Prompt 機制尚未能有效引導模型關注這些特徵。

2-3.3 模型比較與本計畫定位

綜合上述分析，我們可以將 SpeechPrompt 與現有的 SOTA 模型進行詳細的參數與性能對比，如下表 1 所示：

模型架構	範式	可訓練參數量	ASVspoof 2019 LA (EER)
SSL AASIST	Pre-train, Fine-tune	318M	0.21%
MFA-Conformer	Pre-train, Fine-tune	14M	0.72%
SpeechPrompt v2	Prompting Paradigm	0.13M	13.5%

表 2.5: 不同架構之性能與參數比較表。SpeechPrompt 的參數計算基於 Prompt Length $L = 5$ 。

- SSL AASIST：雖然能達到極致的性能表現，但代價是必須微調整個 wav2vec 2.0 XLS-R 巨型模型（約 318M 參數）。這使得訓練成本極高，且每個新任務都需要儲存一個完整的模型副本。

- MFA-Conformer：參考[?]的作法和實驗數據，整體參數量大幅縮減至 14M，同時保持了相當優異的偵測能力。
- SpeechPrompt：該架構在參數效率上達到了極致。根據 Prompt Tuning 的機制，我們僅需訓練 Input Prompts 與 Deep Prompts。假設 Prompt Length 為 5，GSLM (12層, 1024維) 的可訓練參數約為：

$$(2 \times 12 \text{ (Deep layers)} + 1 \text{ (Input layer)}) \times 5 \times 1024 + \text{Verbalizer} \approx 128,000 \approx 0.13M$$

這僅是SpeechPrompt v2 總參數量的 0.8%，SSL AASIST 可訓練參數量的 0.04%，MFA-Conformer 的 1%。

- 研究價值：SpeechPrompt 架構在語意相關任務上展現了顯著的潛力，證明了其作為通用語音處理框架的可行性。然而，其在 ASVspoof 任務上的性能落差 (Underperform)，正是本研究欲探討的核心問題之一。我們將透過實驗探討其性能瓶頸是源於模型架構、特徵損失，還是 Prompt 的設計機制，並試圖提出可能的改進方向。

第三章 實驗結果

3-1. 實驗資料集

為了全面評估模型效能，本實驗採用了以下三個主要的資料集：

1. ASVspoof 系列資料集：

ASVspoof (Automatic Speaker Verification Spoofing and Countermeasures Challenge)

為該領域公認的權威基準。本研究整合了 2019 年與 2021 年的任務場景，以評估模型在傳統攻擊與現代通訊變異下的表現：

- ASVspoof 2019 Logical Access (LA)：作為模型訓練的核心基準，該資料集模擬偽造語音透過通訊系統直接注入的情境。其語料源於 VCTK 資料庫，涵蓋由 17 種文字轉語音 (TTS) 與語音轉換 (VC) 演算法生成的樣本，包含神經聲學模型（如 WaveNet）生成的偽造特徵。
- ASVspoof 2021 Logical Access (LA)：延續 2019 LA 的語料基礎，但進一步引入真實電信網路（VoIP 與 PSTN）傳輸所產生的通道損耗。其經過 A-law、G.722 與 μ -law 等七種不同編解碼器處理，旨在測試模型對於通訊系統所產生的假影（Artifacts）之強健性。
- ASVspoof 2021 DeepFake (DF)：專注於社群媒體與網路傳播場景，偽造樣本由超過 100 種欺騙演算法生成。此子集引入了大量有損壓縮（Lossy compression）處理，包含不同位元速率下的 mp3、m4a 與 ogg 格式，總時數約 454.4 小時，是目前對壓縮抗性要求最高的場景。

2. MLAAD (Multilingual Audio Anti-Spoofing Dataset)：

MLAAD 是一個包含 23 種多語言的大型偽造語音資料集，第二版長度達 160.2 小時，而第八版則達 570.3 小時，所有音訊均以 22.05 kHz 格式輸出。其攻擊樣本生成基於 M-AILABS 語音庫中的真實語音，利用 52 種先進文字轉語音 (TTS) 模型與 19 種不同架構（如 VITS、FastSpeech 等）製作，生成來源廣泛涵蓋 [Coqui.ai](<http://coqui.ai/>) 與 Hugging Face 等開源平台。由於 MLAAD 本身僅包含偽造語音 (Spoof only) 樣本，無法單獨用於訓練二元分類器。因此，在訓練階段，我們參考論文作者建議，將其與此資料集基於的上游全真音訊資料集 M-AILABS 進行混合，以構建完整訓練數據。而在測試階段，我們關注模型對該資料集中偽造語音的檢出率。

3. InTheWild (ITW) :

為了評估模型在實驗室外的泛化能力，本實驗為此採用 InTheWild 資料集作為獨立的跨資料集測試使用。此資料集專門收集自公開的網路來源，用以彌補 ASVspoof 2019 僅基於 VCTK 錄音室語料庫所帶來的局限性。ITW 資料集總長 37.9 小時，精選了 58 位英語名人與政治人物的語音剪輯。其內容被劃分為 17.2 小時的偽造 (fake) 音訊與 20.7 小時的真實 (real) 音訊。偽造音訊是從 219 個公開可用的影片與音訊檔案中分割建構。所有音訊經標準化處理並轉換為 16 kHz 取樣率，平均片段長度約 4.3 秒，整體語料真偽比例維持相較平衡(37.18%)。

Name	Languages	Systems	Utterances	Avg. Dur.	Total Dur.	Spoof Utterances Ratio
ASVspoof19 LA	English	19	121,461	3.25s	109.7hr	89.72%
ASVspoof21 LA	English	13	164,612	2.72s	111.8hr	90.00%
ASVspoof21 DF	English	100+	593,253	3.06s	454.4hr	97.21%
In-The-Wild	English	?	31,779	4.29s	37.9hr	37.18%
M-AILABS	8	0	493,658	7.23s	991.1hr	0.00%
MLAAD v2	23	52	72,000	7.79s	160.2hr	100.00%
MLAAD v8	40	119	243,000	8.45s	570.3hr	100.00%

表 3.1: 實驗資料集統計概覽

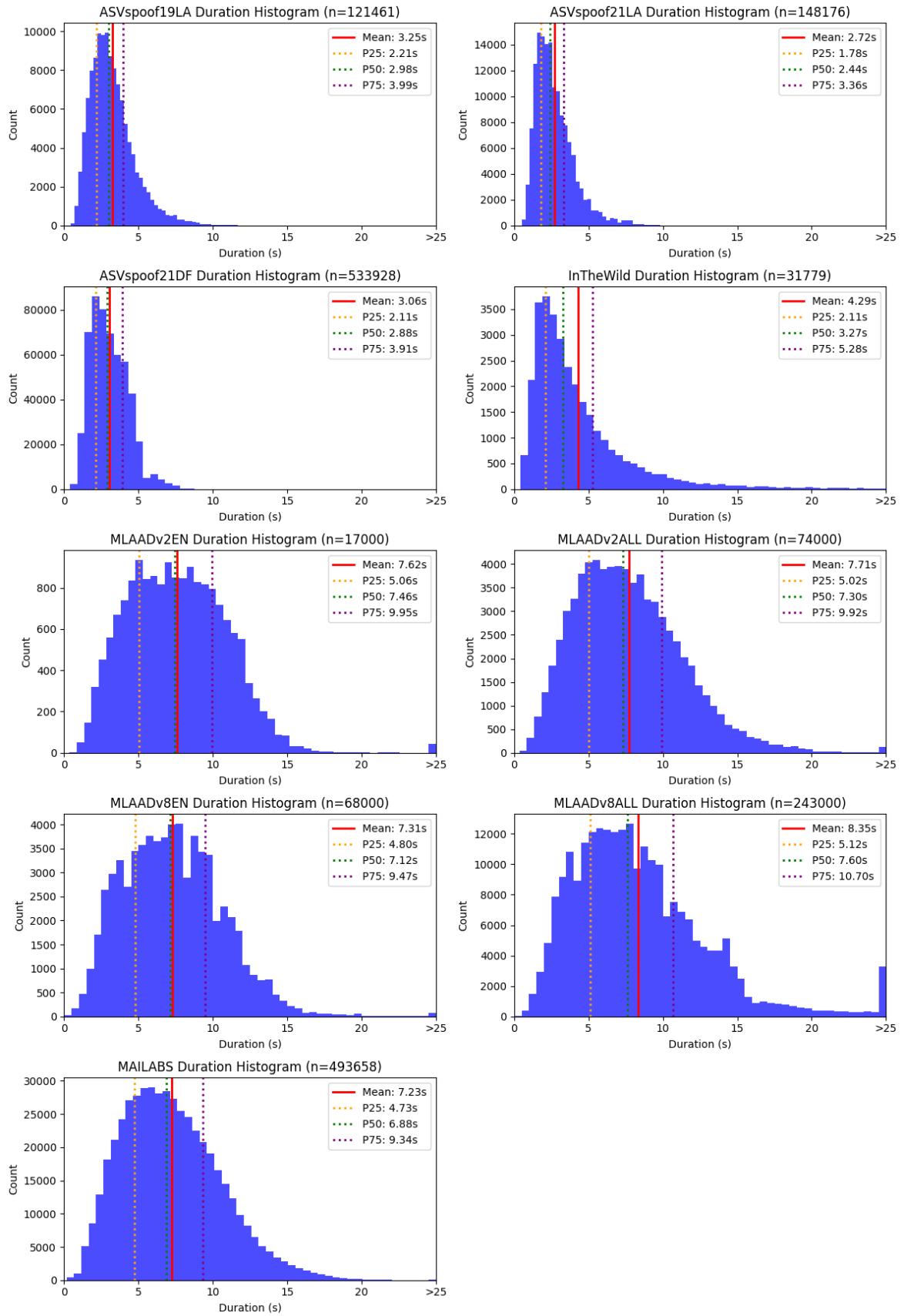


圖 3.1: Caption

3-2. 模型配置與參數設定

3-2.1 硬體環境與實驗控制

所有的實驗均在單張 NVIDIA RTX 3090 (24GB VRAM) 顯卡上進行。為了消除不同模型原始實作中因浮點數精度差異可能帶來的效能影響，並優化記憶體使用效率，本研究對實驗環境進行了以下兩項關鍵的標準化控制：

1. 統一精度 (Unified Precision)：儘管 W2V-AASIST 與 MFA-Conformer 的原始實作採用 FP32 精度，本研究將所有實驗模型的運算精度統一設定為 bfloat16。此舉不僅確保了模型間比較的公平性，更大幅降低了大型模型 (如 SSL-AASIST) 的記憶體佔用，使得我們能夠在單卡環境下使用更大的 Batch Size 進行更穩定的訓練。
2. 統一損失函數 (Unified Loss Function)：為了應對訓練資料中真實與偽造語音可能存在的數量不平衡問題，我們摒棄了 SpeechPrompt 與 MFA-Conformer 原本使用的標準交叉熵 (CE)，將所有模型的損失函數統一固定為加權交叉熵 (Weighted Cross Entropy, WCE)。這與 W2V-AASIST 的原始設定保持一致，確保所有模型在相同的優化目標下進行學習。

3-2.2 模型實作細節

SSL-AASIST (W2V-AASIST)

該模型以原始波形 (Raw Waveform) 作為輸入。由於其前端依賴參數高達 3 億的 wav2vec 2.0 XLSR 模型，為了適應單卡 3090 的記憶體限制並保持訓練穩定性，我們利用 bfloat16 帶來的空間優勢，將 Batch Size 從原始文獻的 14 提升至 40。根據線性

縮放原則，我們相應地調整了學習率，將 Adam 優化器的 Learning Rate 從 10^{-6} 調整至 5×10^{-6} 。輸入長度 (Context Window) 固定為 4 秒。

MFA-Conformer

本研究復現了基於 Conformer 的架構，該模型以 80 維梅爾頻譜圖 (Mel-spectrogram) 作為輸入特徵。實驗中使用 NVIDIA NeMo 提供的預訓練權重 `stt_en_conformer_ctc_small` 進行初始化，其餘超參數與參考文獻保持一致：模型輸入長度設定為 5 秒，Batch Size 設定為 64。優化器採用 Adam，並搭配 Cosine Annealing 排程器 (Scheduler)，包含 Warmup 階段，最高學習率設定為 10^{-3} 。

SpeechPrompt v2

SpeechPrompt 架構採用離散單元 (Discrete Tokens) 作為輸入。由於離散化後的特徵序列長度大幅縮減，除 MLAAD 資料集中極少數超過 100 秒的異常樣本外，模型可將完整的語音長度納入 Context Window 中，無需進行截斷。實驗採用 fairseq 框架進行訓練，由於該框架專為自然語言任務設定特性，會自動根據不同輸入長度動態調整 Batch Size 以盡可能貼近指定的 max token 數量，用以穩定訓練階段的 VRAM 占用，我們採用原始論問的參數進行訓練。在 Prompt 設定方面，我們依據文獻建議，啓用 Deep Prompt 機制，並將 Prompt Length 設定為 5。優化器學習率設定為 5×10^{-3} ，由於 prompt tuning 過程的收斂速度極快，原始論文作法設定 Early Stopping 的 Patience 為 1。

SpeechPrompt 訓練優勢與參數效率分析：儘管 SpeechPrompt 在偵測準確度上未能超越 SOTA 模型，但本計畫的核心目標之一在於探索「參數效率 (Parameter-Efficient)」的解決方案。在可訓練參數量方面，W2V-AASIST 為了達到最佳性能，必須對前端 Wav2Vec 2.0 (XLSR) 進行全參數微調，總可訓練參數量高達

317M。MFA-Conformer (Small) 雖然較為輕量，但仍需訓練約 14.6M 個參數。相比之下，SpeechPrompt 採用 Prompt Tuning 機制，凍結了龐大的 GSLM 骨幹，僅需訓練 Input Prompts、Deep Prompts 與 Verbalizer。在 Prompt Length $X = 5$ 的設定下，其可訓練參數僅為 129,640 (約 0.13M)。這意味著 SpeechPrompt 的參數量僅為 W2V-AASIST 的 0.04%，極大程度地降低了模型儲存與部署的門檻。

在訓練時間成本方面，三者呈現了巨大的差異。(1) W2V-AASIST：由於模型龐大且需處理 Raw Waveform，計算負擔極重。在設定 100 Epoch (針對超大型 MLAAD 資料集調整為 30 Epoch) 的情況下，完整訓練一次需耗時 24 至 36 小時。(2) MFA-Conformer：得益於頻譜特徵的維度縮減，其訓練速度較快。在固定 50 Epoch 的設定下，完整訓練約需 2 小時。(3) SpeechPrompt v2：展現了極致的訓練速度。由於輸入為預先計算好的離散 Token，且需更新的參數極少，模型收斂極快。實驗顯示，其完整訓練過程僅需約 300 秒 (5 分鐘)。綜上所述，SpeechPrompt 展現了以「5 分鐘對比 36 小時」的訓練效率優勢。

3-3. 主要大實驗 Model Performance

評估指標：針對不同的測試情境，本研究採用兩種主要指標。首先，適用於 ASVspoof 與 InTheWild 等包含正負樣本的完整測試集，我們採用 等錯誤率 (Equal Error Rate, EER) 作為主要評估標準。EER 是指當系統的錯誤接受率 (False Acceptance Rate, FAR) 等於錯誤拒絕率 (False Rejection Rate, FRR) 時的值，該對應的值即為 EER，其數學定義如下：

$$EER = FAR(\theta) = FRR(\theta) \quad (3.1)$$

$$\text{where } FAR = \frac{\text{False Acceptances}}{\text{Total Impostor Attempts}}, \quad FRR = \frac{\text{False Rejections}}{\text{Total Genuine Attempts}} \quad (3.2)$$

由此可知，EER 的數值越低，代表系統的辨識與防偽性能越佳。其次，對於 MLAAD 資料集，因其全數為偽造語音（不包含真實語音），無法計算傳統的 EER。因此，我們改採偵測準確率 (Detection Accuracy, ACC)，即計算模型成功將樣本判定為「Spoof」的比例作為準確率指標。

Model	Train Data \ Test on	ASV19 LA	ASV21 LA	ASV21 DF	ITW	MLAAdv2	MLAAdv8
W2V-AASIST (with DA)	ASV19	0.27%	0.87%	5.57%	13.30%	83.24%	71.04%
	ITW	7.14%	10.90%	5.98%	-	55.48%	49.12%
	MLAAdv2	21.29%	30.26%	28.39%	18.77%	-	-
	MLAAdv8	26.85%	30.61%	23.48%	16.03%	-	-
SpeechPrompt v2 (without DA)	ASV19	18.50%	18.23%	28.94%	67.91%	99.65%	99.48%
	ITW	22.23%	25.10%	31.80%	-	98.26%	98.50%
	MLAAdv2	43.92%	44.68%	51.12%	63.31%	-	-
	MLAAdv8	45.10%	46.20%	50.80%	65.10%	-	-
MFA-Conformer (without DA)	ASV19	1.12%	5.67%	16.00%	30.48%	92.61%	88.04%
	ITW	19.93%	26.48%	23.39%	-	37.31%	38.72%
	MLAAdv2	34.90%	41.34%	38.14%	31.34%	-	-
	MLAAdv8	45.27%	45.76%	42.98%	32.75%	-	-

表 3.2: 各模型在不同資料集與訓練設定下之效能表現

本節將整理並對比各模型架構在不同測試集下的實驗結果，詳細數據請見表 3.2。前四個測試集 (ASV19、ASV21 LA、ASV21 DF、ITW) 以等錯誤率 (EER) 為評估指標；後兩個測試集 (MLAAdv2、MLAAdv8) 則以準確率 (Accuracy) 呈現。依據原始設計，我們僅對 W2V-AASIST 採用了其原始論文建議的資料增強 (DA) 策略。

基於標準資料集訓練之表現：當以標準測試集 (ASV19) 進行訓練時，W2V-AASIST 在同領域及跨領域 (ASV21 LA/DF) 測試上均展現出顯著的優勢，EER 分別

低至 0.27%、0.87% 與 5.57%。MFA-Conformer 的表現居次，在 ASV19 上獲得 1.12% 的 EER，但在跨領域的 DF 測試集上升至 16.00%。相對而言，SpeechPrompt v2 在這幾個資料集上的表現明顯落後，EER 皆高於 18%。而在真實場景測試 (InTheWild, ITW) 中，也觀察到類似的趨勢：以 ASV19 訓練的模型，在 ITW 上 W2V-AASIST 能維持 13.30% 的 EER，而 MFA-Conformer 與 SpeechPrompt v2 則分別大幅退步達到 30.48% 與 67.91% 的錯誤率。

基於跨領域資料集訓練之結果：當改以 ITW 作為訓練資料時，所有模型在 ASV 系列上的表現皆普遍不如以 ASV19 訓練的成績。其中，W2V-AASIST 在 ASV19 LA 上維持了 7.14% 的 EER，依然是三個架構中表現最好的；MFA-Conformer 與 SpeechPrompt v2 則分別為 19.93% 與 22.23%，顯示出基礎的反向跨資料集測試依然具備相當的挑戰性。此外，當分別以 MLAAD (v2 或 v8) 進行訓練後去測試其他資料集時，所有架構的 EER 都出現了巨幅的上升，效能全面下滑，顯示出各別依賴於單一資料集的模型特徵極易受限。我們能夠發現這些模型在跨資料集訓練的挑戰上依然存在巨大的模型侷限性。

針對 MLAAD 測試集之表現：在 MLAAD (v2 與 v8) 的評估中，資料呈現出截然不同的反向趨勢。當同樣使用 ASV19 進行訓練時，在 ASV 與 ITW 系列表現最為墊底的 SpeechPrompt v2，卻在 MLAADv2 與 MLAADv8 上達到極高的準確率 (99.65% 與 99.48%)；MFA-Conformer 亦有約 88%-92% 的準確率；相較之下，在其他測試集表現最穩定的 W2V-AASIST 準確率反而僅約 71%-83%。相關現象的深層分析將保留至後續章節與其他消融實驗一起探討。

3-4. Discussion/Analysis

本節針對上述主要實驗中觀察到的現象與模型特性進行深入的消融實驗與分

析。

次要實驗 1：Prompt Tuning 與 Full Fine-Tuning 之比較。為了釐清 SpeechPrompt 性能受限的主因，究竟是源於 Prompt Tuning 方法本身的表達能力不足，還是 GSLM 預訓練模型本身就不具備深偽偵測所需的聲學特徵，我們將測試基準擴展到全模型微調 (Full Fine-Tuning) 並將其視為在此特徵空間下的效能上限 (Upper Bound)。相關數據如表 3.3 所示。

Dataset	Prompt Tuning	Full Fine-Tuning (Upper Bound)
ASVspoof 19 LA	18.50%	17.23%
ASVspoof 21 LA	18.23%	17.60%
ASVspoof 21 DF	28.94%	29.90%

表 3.3: SpeechPrompt v2 在 Prompt Tuning 與 Full Fine-Tuning (Upper Bound) 效能比較

實驗結果顯示，相較於僅使用 Prompt Tuning，全參數量微調 (Full Fine-Tuning) 僅在 LA 任務上帶來小幅度的進步 (EER 由 18.50% 降至 17.23%)，而在深偽難度較高的 DF 任務上甚至出現略微退步 (EER 由 28.94% 上升至 29.90%)。這項結果印證了兩個關鍵洞察：(1) Prompting 的參數效率極高：僅訓練極少量的 Prompt 參數即可達到接近全系統微調的效能水準，並且在面對未知的跨域攻擊時，能藉由避免極度過擬合 (Overfitting) 來維持同等的防禦下限；(2) 瓶頸在於前端特徵：即使開放所有基礎語言模型參數進行微調，也無法取得跨越式的大幅效能進展。這暗示著效能天花板並非源於後端語言模型的學習能力不足，而是前端 S2u (Speech-to-unit) 階段的離散化轉換過程造成了防偽所需的細微連續聲學特徵流失，使得模型在先天上即缺乏辨別真假的關鍵輸入訊號。

次要實驗 2：軟性離散特徵輸入 (Soft Discrete Features)。為了驗證「離散化導致

特徵流失」的假設，並嘗試在不改變模型主體架構的前提下找回遺失的聲學細節，我們提出了一種軟性離散特徵的輸入機制。在原始的 SpeechPrompt 中，HuBERT 輸出的連續特徵會被硬性分配給最接近的中心點。本實驗中，我們改為計算特徵與中心點的相似度分數，並以此分數作為權重，將中心點的 Embedding 進行加權求和作為模型的輸入。我們預期透過這種方式，能在一定程度上保留原始音訊在特徵空間中的相對位置資訊，從而補償量化損失。

次要實驗 3：MFA-Conformer 預訓練模型規模之影響。MFA-Conformer (Small) 在 ASV21 DeepFake (DF) 任務上與 W2V-AASIST 存在顯著落差 (EER 13.43% vs 5.57%)。為了假設這是否源於模型容量不足，我們將前端替換為更大型的 Conformer-Medium 與 Conformer-Large。實驗評估數據如表 3.4 所示：即使將模型規模大幅提升，在 DF 測試集上的 EER 並未出現顯著下降，性能呈現飽和狀態（甚至 Medium 的錯誤率還略高於 Small）。這暗示了單純增加神經網路參數量並無法解決該架構在跨域深偽偵測上的關鍵瓶頸。

Model Size	ASV19 LA	ASV21 LA	ASV21 DF
Conformer-Small	1.25%	4.58%	13.43%
Conformer-Medium	1.13%	3.84%	14.20%
Conformer-Large	1.80%	6.52%	13.31%

表 3.4: MFA-Conformer 不同預訓練模型規模之效能 (EER) 比較

次要實驗 4：MFA-Conformer 後端分類器之改良。我們推測 MFA 原始架構中簡單的統計池化與線性分類器可能無法完全整合 Conformer 抽取的複雜特徵，因此嘗試將 MFA-Conformer 中各層的特徵 (All layer) 導入並替換為 AASIST 後端架構。實驗數據如表 3.5 與原始基礎對齊呈現。

Model Backend	ASV19 LA	ASV21 LA	ASV21 DF
Linear Classifier (Baseline)	1.12%	5.67%	16.00%
AASIST Architecture (using All layer)	0.86%	3.94%	12.71%

表 3.5: MFA-Conformer 不同後端分類器架構之效能 (EER) 比較

結果顯示這項改良帶來了顯著的性能提升：ASV19 LA 的 EER 從 1.12% 下降至 0.86%；ASV21 LA 亦由 5.67% 改善至 3.94%；在處理難度最高的 ASV21 DF 資料集時，表現進步到 12.71%。這證明原先簡易的分類網路限制了 MFA-Conformer 的潛能，透過引入圖神經網路等更強大的特徵融合機制（如 AASIST），確實能更有效率地聚合多層次的前端特徵，並進一步增強防偽系統的穩健性。

第四章 前端應用

4-1. 系統架構與設計目標

為了驗證本研究模型在真實場景中的應用潛力，本研究開發了一套即時音訊深偽偵測系統。該系統不僅提供直觀的視覺化介面，更整合了多種深度學習模型，實現從音訊輸入、特徵分析到結果判定的自動化流程。

4-2. 介面功能與模組說明

本系統採用前後端分離（Decoupled Architecture）架構，以確保系統的擴充性與推論效率：

1. 前端展示層 (Next.js) 採用 Next.js 框架構建。利用其高效的路由管理與組件化開發優勢，實現響應式網頁介面。前端負責處理音訊錄製、檔案上傳與即時波形顯示，並透過 RESTful API 與後端進行異步通訊。
2. 後端推論層 (FastAPI & PyTorch) 後端核心採用 FastAPI 框架，其異步處理能力能有效降低多模型併發推論時的延遲。模型推理部分則由 PyTorch 驅動，整合了本研究所介紹的 ASR 轉錄模型 Whisper、TTS 模型 IndexTTS2、三種深偽檢測模型 MFA-Conformer、W2V-AASIST、SpeechPrompt v2。
3. 快速部署環境 (Docker) 為了應對深度學習環境中複雜的依賴關係（如特定版本的 CUDA、cuDNN、PyTorch 與 FastAPI 依賴庫），本系統全面採用 Docker 容器化技術。這不僅確保了環境的一致性，更極大化地提升了部署效率。

4-2.1 語音輸入

系統提供兩個音訊來源管道，如圖XX左上區塊。使用者可以透過錄音按鈕進行現場錄音，亦可以透過上傳現有音訊檔案。

4-2.2 音訊生成與偽造

平台提供使用者將輸入進的音訊來源當作參考語音透過特定 TTS（如 IndexTTS2）生成偽造語音。生成參考文本可由網頁中上區塊進行輸入，亦可透過 ASR 按鈕將參考音訊透過 Whisper 轉錄成文本進行輸入。生成後的語音將顯示在右上區塊。

4-2.3 偵測核心與效能呈現

當使用者點擊檢測按鈕後，系統會將使用者輸入之音訊以及透過 TTS 生成之音訊送入模型後端進行評測。結果區域以列表方式呈現不同模型的判別數據。

1. 模型標識 (Model ID) 列出參與評測的模型架構。
2. 機率分佈 (Confidence Score) 顯示「真實 (Bonafide)」與「偽造 (Spoof)」的機率百分比。其數值由模型輸出之 Logits 經過 Softmax 轉換所得：

$$P_i = \frac{e^{z_i}}{\sum_{j \in \{B,S\}} e^{z_j}} \quad (4.1)$$

其中 z 為模型最後一層線性層輸出之原始分數 (Logits)， $i \in \{B, S\}$ 分別代表真音與偽造語音類別。

3. 判定門檻 (Threshold) 門檻值 τ 為系統判定真偽的裁決基準。本平台允許針對不同模型設定獨立的門檻值，其設定依據各模型在開發集 (Dev set) 上取得

等錯誤率（EER）時的決策點。若模型輸出之偽造機率 $PS > \tau$ ，系統將判定為「Spoof」並以紅色高亮顯示；反之則判定為「Bonafide」。

4-3. 後端推論流程與模型優化

第五章 結果及討論

5-1.

5-2.

5-2.1

5-3.

5-3.1

5-3.2

第六章 建議與未來展望

6-1.

6-2.

6-2.1

6-3.

6-3.1

6-3.2

參 考 文 獻

- [1] Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Göknar, Iulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and Julian Weber. XTTS: a Massively Multilingual Zero-Shot Text-to-Speech Model. In Interspeech 2024, pages 4978–4982, 2024.
- [2] Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, sheng zhao, and Tie-Yan Liu. Adaspeech: Adaptive text to speech for custom voice. In International Conference on Learning Representations, 2021.
- [3] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching, 2024.
- [4] Junyoung Chung, Çaglar Gülcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. ArXiv, abs/1412.3555, 2014.
- [5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1126–1135. PMLR, 06–11 Aug 2017.
- [6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. Journal of Machine Learning Research, 17(59):1–35, 2016.

- [7] Xavi Gonzalvo, Siamak Tazari, Chun an Chan, Markus Becker, Alexander Gutkin, and Hanna Silen. Recent advances in google real-time hmm-driven unit selection synthesizer. In Interspeech 2016, pages 2238–2242, 2016.
- [8] Sung-Feng Huang, Chyi-Jiunn Lin, Da-Rong Liu, Yi-Chen Chen, and Hung-yi Lee. Meta-tts: Meta-learning for few-shot speaker adaptive text-to-speech. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30:1558–1571, 2022.
- [9] Shuang Leng, Yue Chen, Zhaozheng Liu, Jiaqi Li, Zhuhong Yuan, Rongye Zhang, Kaitao He, Xinfu Guo, Feijuan Chen, Xianghao Song, et al. Fish-speech: Leveraging large language models for advanced multilingual text-to-speech synthesis, 2024.
- [10] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In International Conference on Learning Representations, 2020.
- [11] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fast-speech 2: Fast and high-quality end-to-end text to speech. In International Conference on Learning Representations, 2021.
- [12] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. ArXiv, abs/1505.00387, 2015.
- [13] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems (NeurIPS), pages 3104–3112, 2014.
- [14] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaity, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis

- Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. In Interspeech 2017, pages 4006–4010, 2017.
- [15] Heiga Zen, Yannis Agiomyrgiannakis, Niels Egberts, Fergus Henderson, and Przemysław Szczępaniak. Fast, compact, and high quality lstm-rnn based statistical parametric speech synthesizers for mobile devices. In Proc. Interspeech, pages 662–666, 2016.
- [16] Siyi Zhou, Yiquan Zhou, Yi He, Xun Zhou, Jinchao Wang, Wei Deng, and Jingchen Shu. Indextts2: A breakthrough in emotionally expressive and duration-controlled auto-regressive zero-shot text-to-speech, 2025.